## LPD-17 Self-Protection

H.H. Mok and R.J. Futato
*Tactical Electronic Warfare Division*

**Introduction:** Ensuring operational readiness under combat conditions is important for the development of all new weapons systems. The U.S. Congress recently mandated that all new weapons systems be subjected to live fire conditions prior to deployment to assess their vulnerability and survivability. For existing Navy surface combatants, such evaluation is not feasible using field trials because of the prohibitive cost of testing in a scientifically meaningful way within acceptable risk levels. For systems still in the design phase (the optimal time to discover and correct system deficiencies), actual field trials are outright impossible. High-fidelity, physics-based digital simulation with a strong validation and verification process is therefore being accepted as a viable alternative. The CRUISE_Missiles electronic warfare simulation and Radar Target Signature (RTS) ship signature prediction model are proving instrumental in supporting self-protection evaluations for a number of modern ship platforms, including most recently the LPD-17 Amphibious Transport Dock Program. Specifically, the effort is directed toward the Live Fire Test & Evaluation (LFT&E) Hit-Point assessment, Probability of Raid Annihilation (PRA), and the effectiveness of proposed Ship Self-Defense Systems (SSDS).

**CRUISE_Missiles Modeling Overview:** For more than a decade, the Advanced Techniques Branch of the Tactical Electronic Warfare Division (TEWD) has fostered the evolution of a sophisticated digital electronic warfare (EW) simulation model, CRUISE_Missiles. This model focuses on supporting the study and design of improved weapons systems and countermeasures (CM) for defending Navy ships against antiship cruise missiles (ASCM) under combat conditions. Breakthroughs in modeling technology by NRL's Tactical Electronic Warfare and Radar Divisions have augmented CRUISE_Missiles with an inventory of modern ship representations that create true, spatially extended signatures directly traceable to ship architecture.[1,2] The coupling of these signature representations with the powerful numerical algorithms optimized for supercomputing within CRUISE_Missiles permits accurate simulation of the complex envelope signal processing of fundamentally important ASCM/ship/EW interactions, on a pulse-by-pulse basis, at the actual pulse repetition frequency (PRF) of the missile radar. Currently, CRUISE_Missiles is able to handle many-on-many engagements of modern ASCMs and low-observable ship designs, realistic ship motion models with sea state and environment influences, target discrimination, passive and active CM models, and combined CM tactics. The model validation and verification approach mirrors physical experiment and field measurement. The software development and maintenance processes follow Defense Modeling and Simulation Office (DMSO) guidelines.

**LPD-17 ASCM Studies:** Ship signature characteristics arising from structural design decisions can have a major impact on defensibility and survivability against ASCM attacks. It is therefore desirable to include ASCM evaluations early in the design process when significant problems can be more easily identified and corrected. For LPD-17, a CRUISE_Missiles ship model is derived for each major design revision, and a series of hit point and softkill (threat deception) effectiveness studies are performed. Parallel studies are performed both for radar-guided (RF) and infrared-guided (IR) ASCM threats using similar approaches, described below for the RF regime.

Each RF CRUISE_Missiles LPD-17 model is derived from a parent RTS model developed and maintained by the Naval Surface Warfare Center, Carderock Division (NSWCCD). The full RTS representation is generated directly from the computer-aided design (CAD) model of the ship using utilities that transform geometric primitives into RF scattering primitives. The resultant RTS model thus comprises scattering primitives for all major structural components, typically totaling more than 1 million scatterers (see Fig. 1). To reduce computation time, the CRUISE_Missiles model is derived by choosing from this full model only those scattering primitives of sufficient magnitude to be relevant to internal ASCM processes.[1] This resulting model can be processed by the simulated radar signal chain at true PRF rates while maintaining signature accuracy in terms of both total radar cross section and downrange distribution. Moreover, it integrates first-principle signature characterizations with full PRF ASCM models in a manner directly traceable to the underlying ship structure.

Evaluation of surface ship operational readiness must consider defensibility against ASCM attack as well as survivability in the event of defense failure. Series of CRUISE_Missiles runs are therefore made for each LPD-17 design revision in both the undefended (baseline) and defended configuration. Runs for the undefended configuration are made first to characterize the inherent design susceptibility to ASCM attack. Tracking behavior extracted from these

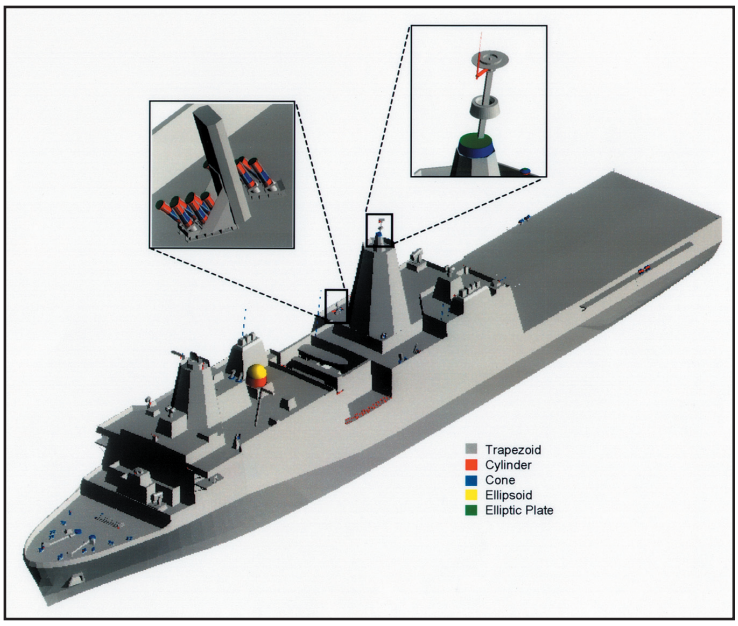


**FIGURE 1**
RTS ship signature model representation for the LPD-17.

runs is used to differentiate signature-induced hits from chance ballistic hits, to identify dominant structural features, and to derive preliminary self-defense tactics from established Fleet doctrine. Runs are then repeated using proposed decoys and preliminary deployment tactics to characterize softkill effectiveness. Figure 2 illustrates the results from an idealized run involving two-round seduction chaff defense. Detailed signal-level analyses (supported by high-fidelity physics-based simulations) are performed on each such run to understand the mechanism underlying the observed result. Considerations such as the initial ASCM tracking behavior prior to decoy deployment and the nature of the internal threat decision process when presented with the combined signatures are investigated for potential use in ship and tactics refinement.

Vulnerability assessments are made using both the baseline and defended data sets. In both cases, statistically significant hit-point distributions are generated to characterize the threat location and velocity at detonation as a function of angle-of-attack for identified potential threats and relevant environmental conditions (Fig. 3). Each CRUISE_Missiles run fully characterizes the ASCM flight profile during guided flight toward the selected target, ending with the location and velocity of the threat at the onset of ballistic flight. For runs resulting in ship hits, the final velocity vectors are extrapolated to the surface of the full RTS model (which contains a complete description of the ship surface topography) to determine the location and velocity of the ASCM threat at detonation. The combination of accurate spatially extended signature representations with detailed ASCM threat models provides high-confidence estimates of RF-guided hit points for use in ship survivability assessments. The hit point distributions ultimately are used as input to the Ship Vulnerability Model (SVM) maintained by NSWCCD to further assess the damage of the missile impact and analyze the survivability of the combatant.

**Future Directions:** Vulnerability and susceptibility of Navy surface ships is strongly dependent on the success of defending against ASCM attack. CRUISE_Missiles has played and will continue to play a major role in improving survivability against ASCM threats via signature control and self-defense measures for LPD-17 and other emerging surface designs. Anticipated contributions to these programs include the development of optimal design modifications and treatment plans based on ASCM threat performance measures, the prediction of ASCM flight trajectories for use in the evaluation of hardkill (threat interception) effectiveness as part of PRA assessments, and the evaluation of coordinated hardkill/softkill tactics.

[Sponsored by NAVSEA]

**Reference**

[1] A.J. Goldberg and R.J. Futato, "CRUISE_Missiles Electronic Warfare Simulation," *1993 NRL Review*, pp. 123-126.
[2] M.A. Busse and D.A. Zolnick, "Reduction of Ship Radar Cross Section Using Measured and Calculated Signature Data," *1998 NRL Review*, pp. 173-175.

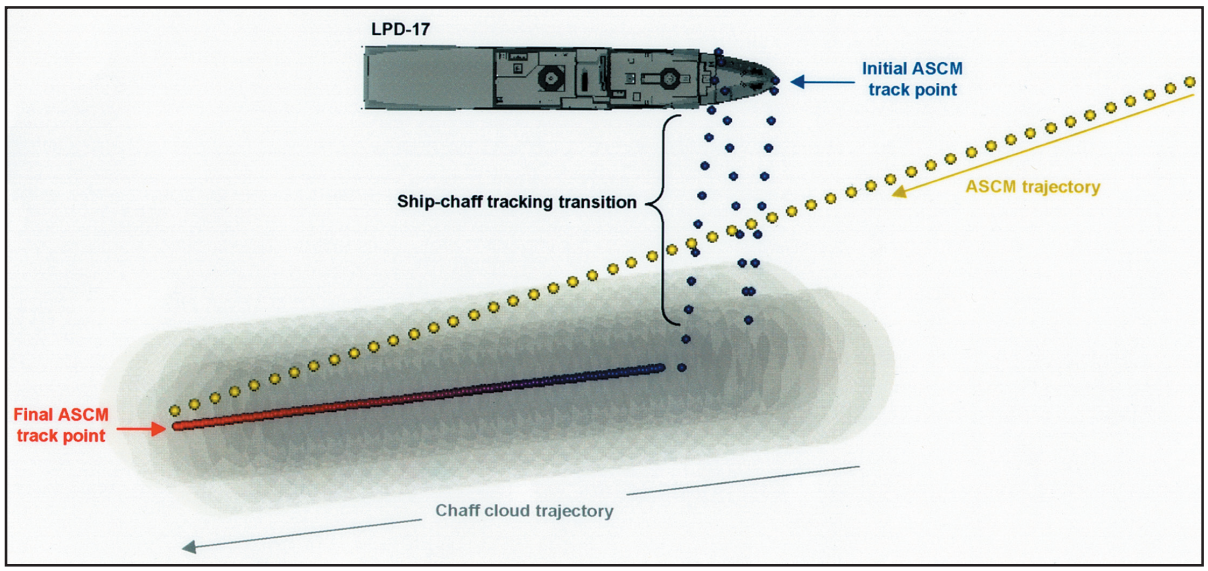


**FIGURE 2**
Hypothetical ASCM/LPD-17 engagement involving two-round seduction chaff defense.

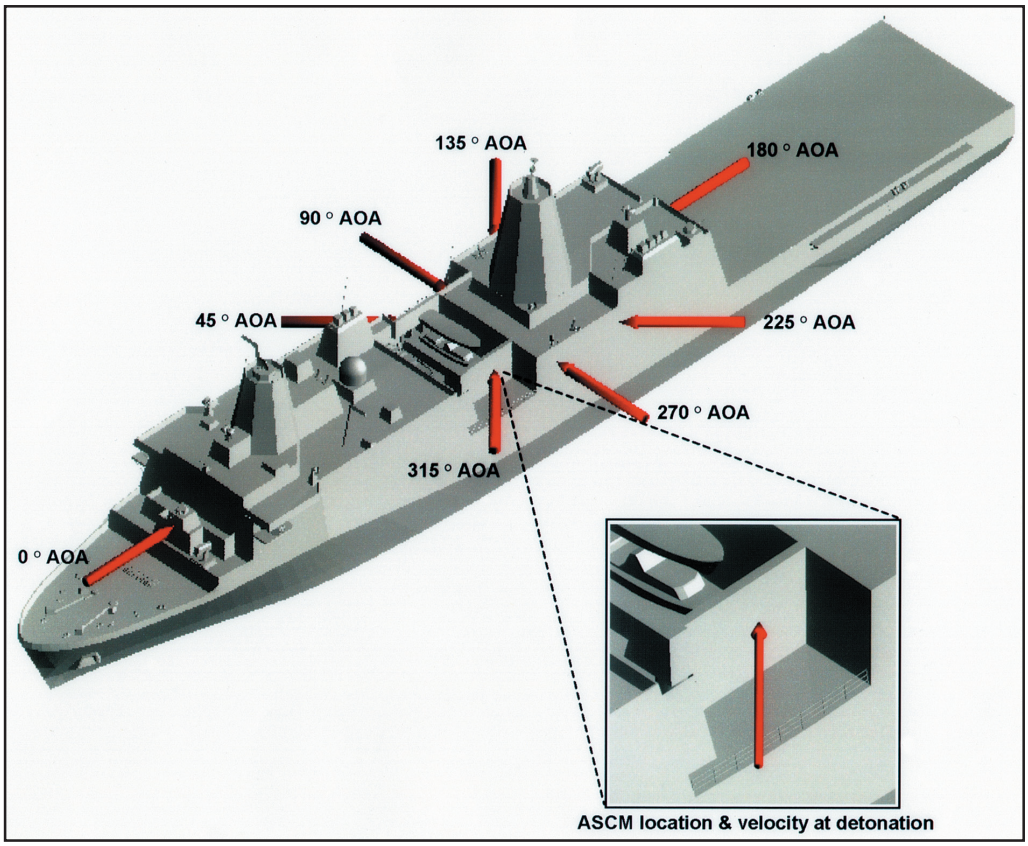


**FIGURE 3**
Hypothetical ASCM hit-points.

## An Advanced Simulation Tool for Damage Assessment

K. Kailasanath, D. Schwer, and G. Patnaik
*Laboratory for Computational Physics & Fluid Dynamics*

**Introduction:** One component of improved ship design is the development of technologies that will reduce a ship's vulnerability to weapon impact. Quick identification of the location of weapon impact and an assessment of the potential damage that may be caused by it can significantly reduce vulnerability by providing protection for onboard personnel, weapons, and systems. The planned reduction in manning of future ships and submarines also requires the development of improved tools for automated response to fires and explosions. Such damage containment decision aids and future automated damage control systems need the input from physics-based models and simulations to be realistic and reliable.

The objective of our research effort is to develop a computational tool that can be used to assess the damage to the interior compartment of a surface ship under different war-fighting and peacetime scenarios. For example, such a tool should be capable of simulating the dispersal of fire suppressants and their interaction with a fire in the complex geometry that is typical of the interior of a surface ship.

**Computational Strategy:** When this project started in FY99, there was no capability to efficiently simulate the detailed flow field inside the complex geometry that is typical of a ship compartment. Zone models that might provide global answers, such as heat transfer from one compartment to the next, do not have the resolution to capture the fluid dynamics of the dispersal of fire suppressants injected from a nozzle into a compartment. These models also cannot simulate the interactions between the fire suppressant and a fire in the enclosure. Detailed numerical simulation models have the ability to capture the local interactions between a fire suppressant (such as water-mist) and a fire. However, the numerical grid resolution that is required to correctly capture the local interactions with a fire make it prohibitively expensive to extend this approach directly to simulate fire suppression in a large compartment. Furthermore, such models do not have the capability to represent the complex geometrical details that are typical of a ship compartment and cannot cost-effectively compute flows evolving over several minutes. A new approach was needed to compute the complex flows over long durations.

Multidomain computational techniques have been adopted to combine flows with different levels of complexity evolving in different regions (or domains) of the system. By using coarser grids and larger timesteps wherever the flow is evolving slowly and smoothly, significant gains in computational time has been made without loss of detail. In combination with this multidomain technique, an efficient approach has also been developed to represent the geometric complexity of the interior of a ship.

**Smoke Spread through a Ship:** The developed simulation tool is used to demonstrate smoke spread through multiple compartments of the ex-USS *Shadwell*, the Navy's damage-control and fire-fighting research ship.[1] Figure 4 shows details of the interior of the ship that are represented in the model. There are 13 compartments with 9 doors and 8 hatches. In Fig. 5, the smoke spread from an uncontained fire in the laundry room is shown in terms

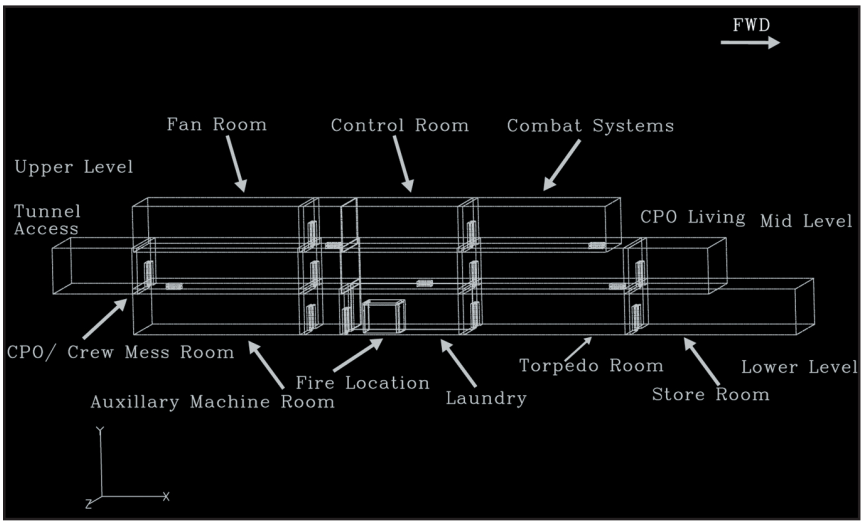


**FIGURE 4**
A schematic of the different compartments of the ex-USS *Shadwell* represented in the numerical simulations.

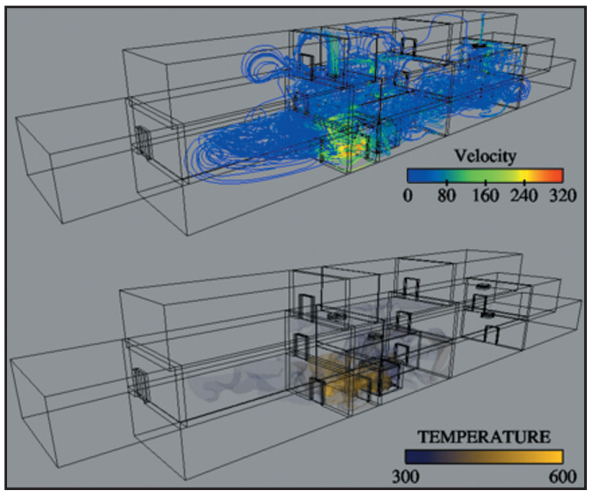


**FIGURE 5**
The simulated smoke spread within the ex-USS *Shadwell* shown using velocity (in cm/s) and temperature (in K) distributions.

of temperature and velocity distributions. In Fig. 6, details of the local flow field in the laundry room and the laundry-room passageway are shown to highlight the fact that local details, where needed, have not been sacrificed in the simulation tool.

Parametric studies have also been carried out to determine the effects of varying factors such as droplet diameter, mist-density, injection velocity, and nozzle location on the suppression of a fire in one of the compartments. In addition, the effect of blockages within the compartment on the entrainment of water-mist and fire suppression has also been investigated.[2]

**Significance:** The damage to the interior compartment of a surface ship under different war-fighting and peacetime scenarios can be simulated using the developed tool. Effectively using this tool in the design process will lead to a ship design that is improved in active and passive fire protection. The development of such a tool can also be thought of as a first step toward better-automated response to fire scenarios and the consequent reduction in manning requirements.

**Acknowledgments:** The authors acknowledge the contributions to this research effort by Dr. Kuldeep Prasad, while working at NRL.

[Sponsored by ONR]

**References**

[1] K. Prasad, G. Patnaik, and K. Kailasanath, "Advanced Simulation Tool for Improved Damage Assessment. 1—A Multiblock Technique for Simulating Fire and Smoke Spread in Large Complex Enclosures," NRL/MR/6410-00-8428, February 2000.

[2] D.A. Schwer, G. Patnaik, and K. Kailasanath, "Effect of Obstructions on Fire-Suppression for Large Enclosed Fires," *Proceedings of the Fall Technical Meeting of the Combustion Institute*, December 2001.

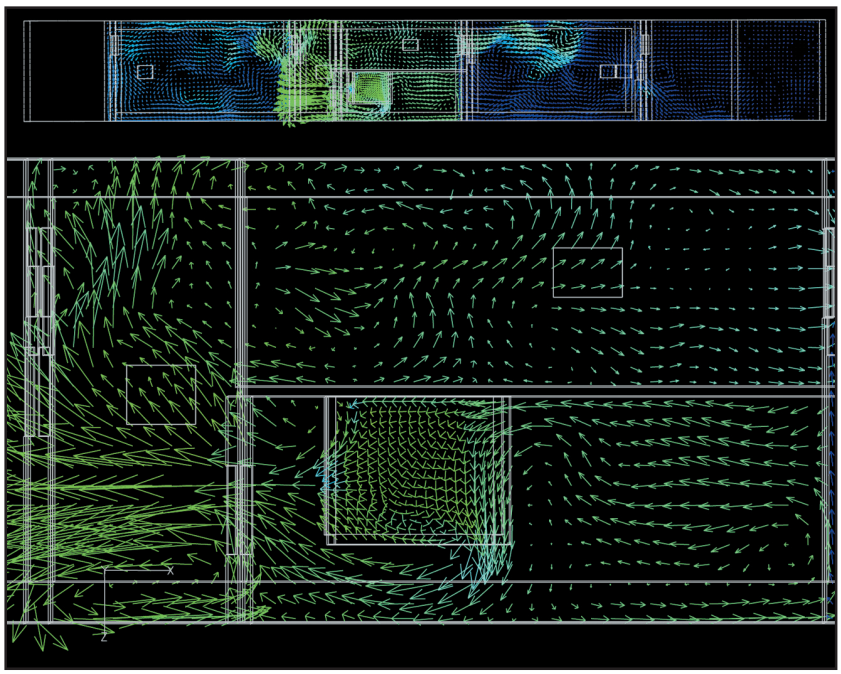

**FIGURE 6**
A detailed view using velocity vectors to illustrate the flow field in the laundry room and the adjacent passageway.

# Embedded Processor Analysis/ Simulation Tools

M.A. Harball
*Tactical Electronic Warfare Division*

N. Bond
*CACI, Inc.*

**Introduction:** The benefits of using embedded processors in larger systems are well documented—cheaper, faster processing, improved reliability, enhanced flexibility, etc. However, the embedded device also complicates development, analysis, testing, and validation. Functions developed in software create a new level of abstraction for system design and analysis. System operation with an embedded machine involves the interaction of the system hardware, the embedded processing hardware, and the embedded processing software. Validating the composite system requires demonstrating that all of these levels function and interact in real time as the designer intended. Our objective was to investigate and develop a flexible embedded system emulation/modeling architecture to provide the ability to rapidly and accurately model a number of standard, off-the-shelf embedded devices running in a user-defined simulated hardware environment. This facilitates rapid prototyping and simultaneous development of hardware and software.

Analysis of embedded systems software operation involves emulating or modeling the processor operation within the target system hardware. This is typically done using a commercial emulation package that provides the user with the ability to run or step through the program, examining and modifying processor and program registers and flags. This process provides accurate modeling and analysis of the interactions between the system hardware and software, but it also requires that the hardware be complete—the design and implementation finalized—to validate the entire system. Modeling fidelity is an important issue in this approach.

**Our Approach:** Our approach, developed under contract by CACI, Inc., replaces the target system hardware with a simulation contained in in-system programmable (ISP) field programmable gate arrays (FPGAs). An actual processor is substituted for the emulator stand-in. With both emulation logic and system logic functions provided within the FPGAs, the designer has the ability to monitor operation of the entire embedded system. The objective is an emulation environment that is flexible and able to support a variety of embedded machine designs. The first goal is to provide support capability for typical 8-bit and 16-bit microprocessors and 16-bit digital signal processor architectures. The embedded emulation logic design and interface is specific to each processor and in some cases to each model. However, once this interface is established, the logic analysis and code development tools are available from within the emulation/modeling tool set.

With a sample of the actual embedded machine running the code, modeling and validation issues are greatly simplified. Use of the processor as an emulator requires a set of hardware control tools to provide the usual emulation features and connect them to a host machine. These are coded into a second FPGA. This first design runs in a commercial off-the-shelf (COTS) PC so that commercial FPGA development tools can be run on the same host. However, the ANSI-VITA-4 compatible industry package (IP) format allows the emulator hardware to be readily adapted to any host computer supported by a range of commercially available adaptor boards.

Figure 7 shows a diagram of our emulation architecture. All key hardware elements are contained in the IP card, which appears in Figs. 8 and 9 (top and bottom views). The processor itself is mounted on a separate card that plugs "piggy back" onto the IP card, so each processor requires its own processor adapter card (PAC) with the appropriate emulation control hardware configuration downloaded into the FPGA. Expanding to a second processor, e.g., adding a DSP chip for dedicated functions, is a matter of providing a second IP board, emulator control configuration, and PAC. A separate dedicated bus is available for direct connections between IP cards, providing the capability to mix and match processor configurations within a single host. The integrated package also provides a means to incorporate (via the FPGA) hardware-based logic analysis tools to monitor and modify the embedded code, the processor operation, and the modeled system hardware components—all in a very small physical package that can be installed in a desktop workstation such as a PC.

**Conclusions:** This approach has several important advantages. Since the system hardware model is loaded into an FPGA, it can be developed rapidly using standard commercial tools and modified easily as faults or improvements are identified. Software development and accurate, detailed functional analyses can be initiated with limited definition of the final system hardware. Finally, except for the core IP hardware and PAC cards, the entire system design (including the analysis tools and external hardware simulation) is user-definable via software and can be downloaded to the emulator at startup.

[Sponsored by ONR]

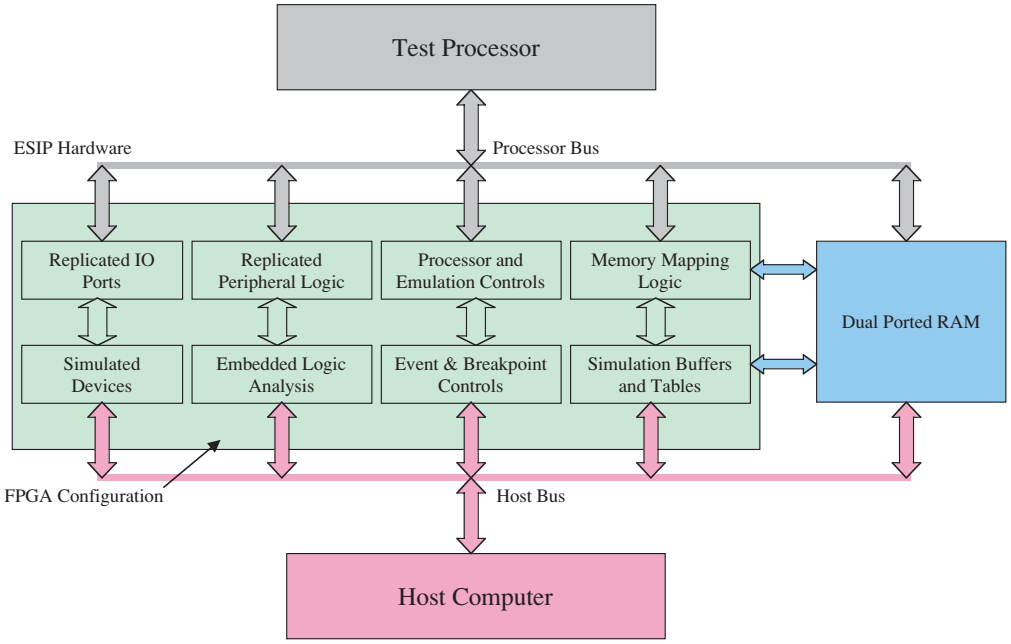


**FIGURE 7**
Embedded System Emulation Architecture.

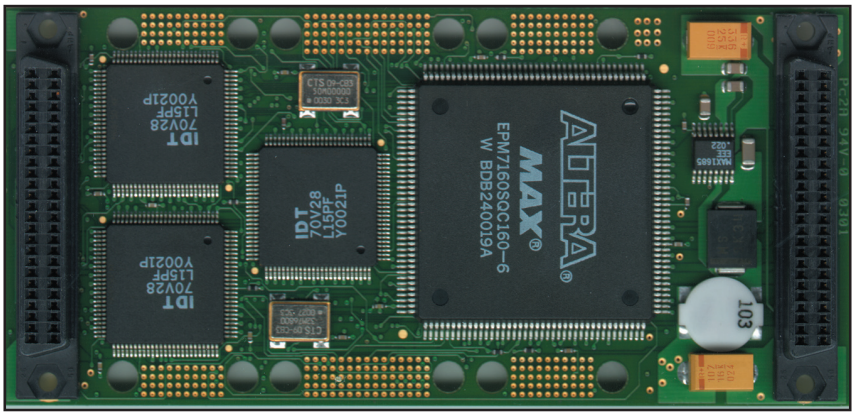


**FIGURE 8**
IP memory and host interface hardware.

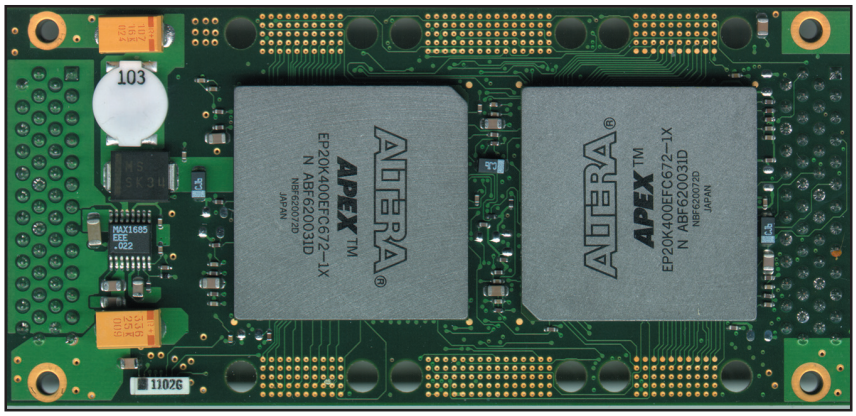


**FIGURE 9**
IP FPGA and processor interface hardware.